In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 12, lines 6 to 30 as follows:

--Multiprocessor integrated circuit 100 is constructed to provide a high rate of data transfer between processors and memory using plural independent parallel data transfers. Crossbar 50 enables these data transfers. Each digital image/graphics processor 71, 72, 73 and 74 has three memory ports that may operate simultaneously each cycle. An instruction port (I) may fetch 64 bit data words from the corresponding instruction cache. A local data port (L) may read a 32 bit data word from or write a 32 bit data word into the data memories or the parameter memory corresponding to that digital image/graphics processor. A global data port (G) may read a 32 bit data word from or write a 32 bit data word into any of the data memories or the parameter memories or random access memory 20. Master Processor 60 includes two memory ports. An instruction port (I) may fetch a 32 bit instruction word from either of the instruction caches 11 and 12. A data port © (C) may read a 32 bit data word from or write a 32 bit data word into data caches 13 or 14, parameter memory 15 of random access memory 10 or any of the data memories, the parameter memories or random access memory 20. Transfer controller 80 can access any of the sections of random access memory 10 or 20 via data port (C). Thus fifteen parallel memory accesses may be requested at any single memory cycle. Random access memories 10 and 20 are divided into 25 memories in order to support so many parallel accesses.--

Rewrite the paragraph at page 27, lines 1 to 4 (Table 1) as follows:

--

Symbol	Operation
~	bit wise NOT
&	bit wise AND
~~|~~ |	bit wise OR
^	bit wise exclusive OR
%	mask generation
%!	modified mask generation
>>	shift right
~~||~~ ||	parallel operation

Table 1--

Rewrite the paragraph at page 46, line 25 to page 47, line 14 as follows:

--The "E" bit (bit 14) designates an explicit multiple carry-in. This bit permits the carry-in to be specified at run time by the input to the C-port of arithmetic logic unit 230. If both the "A" bit and the "E" bit are "1" and the "FMOD" field does not designate the cin function, then the effects of the "S", "I" and "C" bits are annulled. The carry input to each section during multiple arithmetic is taken as the exclusive OR of the least significant bit of the corresponding section input to the C-port and the function signal F0. If multiple arithmetic is not selected the single carry-in to bit 0 of arithmetic logic unit 230 is the exclusive OR of the least significant bit (bit 0) the input to the C-port and the function signal F0. This is particularly useful for performing multiple arithmetic in which differing functions are performed in different sections. One extended arithmetic logic unit operation corresponds to ~~(A^B)&C ~~ ~~(A^~B)&C~~ (A^B)&C | (A^~B)&C. Using a mask for the C-port input, a section

with all "0's" produces addition with the proper carry-in of "0" and a section of all "1's" produces subtraction with the proper carry-in of "1".--

Rewrite the paragraph at page 51, line 29 to page 52, line 21 as follows:

--The output of multiplier 220 supplies the input of product left shifter 224. Product left shifter 224 can provide a controllable left shift of 3, 2, 1 or 0 bits. The output of multiply shift multiplexer MSmux 225 controls the amount of left shift of product left shifter 224. Multiply shift multiplexer MSmux 225 selects either bits 9-8 from the "DMS" field of data register D0 or all zeroes depending on the instruction word. In the preferred embodiment, multiply shift multiplexer MSmux 225 selects the "0" input for the instructions ~~MPYx ° ADD and MPYx ° SUB~~ MPYx ‖ ADD and MPYx ‖ SUB. These instructions combine signed or unsigned multiplication with addition or subtractions using arithmetic logical unit 230. In the preferred embodiment, multiply shift multiplexer MSmux 225 selects bits 9-8 of data register D0 for the instructions ~~MPYx ° EALUx~~ MPYx ‖ EALUx. These instructions combine signed or unsigned multiplication with one of two types of extended arithmetic logic unit instructions using arithmetic logic unit 230. The operation of data unit 110 when executing these instructions will be further described below. Product left shifter 224 discards the most significant bits shifted out and fills the least significant bits shifted in with zeros. Product left shifter 224 supplies a 32 bit output connected to a second input of multiplexer Rmux 221.--

Rewrite the paragraph at page 84, lines 7 to 19 as follows:

-- 1a. p1 = a1 * x1

```
2a.         p2 = a2 * x2
~~2b.  °    q1 = b1 + p1>>8~~
2b.   ||    q1 = b1 + p1>>8

3a.         r1 = q1 * x1
~~3b.  °    q2 = b2 + p2>>8~~
3b.   ||    q2 = b2 + p2>>8

4a.         r1 = q2 * x2
~~4b.  °    s1 = c1 + r1>>8~~
4b.   ||    s1 = c1 + r1>>8

5a.         y1 = s1 * x1
~~5b.  °    s2 = c2 + r2>>8~~
5b.   ||    s2 = c2 + r2>>8--
```